amend elements of bottom-most portion as shown; and
add element numbering as shown.

Insert NEW figure -- FIG. 15d --
Insert NEW figure -- FIG. 15e --
Insert NEW figure -- FIG. 15f --
Insert NEW figure -- FIG. 15g --
Insert NEW figure -- FIG. 17 --.

IN THE SPECIFICATION:

The paragraph beginning at **page 2, line 22** has been amended as follows:

"One problem is that the basis for nearly all interfacing remains that of traditional graphical user interfaces ("GUIs"). Compared with typed "command-lines," GUIs offer a more intuitive pointer-based mechanism for stepping programs through their operational steps. Traditional GUIs form separate graphic depictions according to displayable functional divisions (e.g. windows) within which a user finds and selects *data* indicators; the user can then select a *tool* to affect the already-selected data. Each mouse movement and "click" is simple and encapsulated, and a simple feedback indicator reflects that the function has been initiated (e.g. menu item graying, button "pushing", and so on). Unfortunately, using such a GUI to find items, enter data and execute mouse motions can be time consuming, confusing and distracting."

The paragraph beginning at **page 4, line 19** has been amended as follows:

"Aspects of the invention provide for interfacing one or more users or groups of users with one or more machines or groups of machines in a manner adaptable to conventional and non-conventional control and data entry capabilities, and that can be used alone and in conjunction with other interfacing elements, approaches and operabilities. Complete interfaces, command sets, commands, feedback, and new environments and operations are also enabled in a modifiable, extensible, broadly applicable and portable manner that is capable of exploiting flexibility, suggestibility and other aspects of human expression. New interfacing and

programming tools ~~also~~ can also be used to replace, extend and/or be superimposed with conventional/non-conventional interface elements, capabilities and/or other more useful "expressive" and/or "event-based" interfacing."

The paragraph beginning at **page 5, line 15** has been amended as follows:

"Embodiments further provide "conversational factors" and other aspects according to which commands can be structured, populated with verbiage and rendered operable in a replacing, independent, complimentary, integrateable and/or superimposed manner. For example, a user's ongoing expectations can be met/guided by forming conversant structures and populating the structures with verbiage to apply consistent "rhythmic flow" within a general or specific conversant context (e.g. working with lists); rhythmic flow can also be made to vary in a determinable manner according to factors such as a new, sub or intermittent context, and/or mentally combinable verbiage portions for providing varying degrees of implicit or explicit specificities of user, subject, object, or action/tool variations. Such structures/verbiage can further enable direct/indirect and explicit/implicit data input (e.g. dictation) as well as control, e.g. via generally applicable, specific or automatically relatable "designations"."

The BRIEF DESCRIPTION OF THE DRAWINGS, which begins on **page 8, line 3** has been amended as follows:

"FIG. 1 is a flow diagram illustrating a conversance-enabled system according to an embodiment of the invention;

FIG. 2 is a block diagram illustrating a processing system according to an embodiment of the invention;

FIG. 3a illustrates an assistant scenario according to an embodiment of the invention;

FIG. 3b is a flow diagram illustrating an application of conversant aspects and interfacing, including contexts, tasks, specificity and enhancement, according to an embodiment of the invention;

FIG. 3c is a flow diagram illustrating a further application of conversant aspects and interfacing to local/remote real or virtual machines, according to an embodiment of the

invention;

FIG. 4a is a flow diagram illustrating a command creator according to an embodiment of the invention;

FIG. 4b is a block diagram illustrating an I/O analyzer according to an embodiment of the invention;

FIG. 4c is a flow diagram illustrating a distributed command creator according to an embodiment of the invention;

FIG. ~~6b~~4d is a flow diagram illustrating a ~~further~~ command executor according to an embodiment of the invention;

FIG. 5a is a block diagram illustrating a command structure according to an embodiment of the invention;

FIG. 5b illustrates a command group according to an embodiment of the invention;

FIG. 6a is a flow diagram illustrating a command executor according to an embodiment of the invention;

~~FIG. 6b is a flow diagram illustrating a further command executor according to an embodiment of the invention;~~

FIG. 7a is a block diagram illustrating an input engine according to an embodiment of the invention;

FIG. 7b is a block diagram illustrating a command engine according to an embodiment of the invention;

FIG. 7c is a block diagram illustrating an application support engine according to an embodiment of the invention;\

FIG. 7d is a block diagram illustrating a designation engine according to an embodiment of the invention;

FIG. 7e is a block diagram illustrating an enhancement engine according to an embodiment of the invention;

FIG. 8a is a block diagram illustrating a user engine according to an embodiment of the invention;

FIG. 8b is a block diagram illustrating a use engine according to an embodiment of the invention;

FIG. 8c is a block diagram illustrating a security engine according to an

embodiment of the invention;

FIG. 8d is a block diagram illustrating a reliability engine according to an embodiment of the invention;

FIG. 8e is a block diagram illustrating a conversance engine according to an embodiment of the invention;

FIG. 8f is a block diagram illustrating a history engine according to an embodiment of the invention; ~~and~~

FIG. 8g is a block diagram illustrating an information retriever according to an embodiment of the invention~~.~~;

FIG. 9a is a flow diagram illustrating a single machine interfacing system according to an embodiment of the invention;

FIG. 9b is a flow diagram illustrating a distributed interfacing system according to an embodiment of the invention;

FIG. 9c is a flow diagram illustrating an interfacing system including gesturing and non-gesturing interfacing, and multiple speech programs, according to an embodiment of the invention;

FIG. 9d is a flow diagram illustrating an interfacing system including information recording and including multiple recognition engines and associate-able voice commands and operational code for interfacing with one or more applications, according to an embodiment of the invention;

FIG. 9e is a flow diagram illustrating an interfacing system employing local speech processing and coordinating such processing with distributable remote parameters, application handling or other operations, according to an embodiment of the invention;

FIG. 10a is a flow diagram illustrating an example of how an interfacing according to an embodiment of the invention may provide for handling an underlying application as well as data map or other intra/extra host information storage/retrieval, non-speech interfacing, network (e.g., Internet) based interaction and recording of information corresponding thereto;

FIG. 10b is a flow diagram illustrating how interfacing according to an embodiment of the invention may be used by multiple users using different devices for conducting one

or more group/workgroup interactions corresponding to one or more applications, as well as extra-group/workgroup tasks;

FIG. 10c is a flow diagram illustrating conversion of conversant interfacing information according to task, context or inter-context parameters, according to an embodiment of the invention;

FIG. 10d is a flow diagram illustrating how a target machine may be analyzed and provided with interfacing or control capabilities according to an embodiment of the invention;

FIG. 11a illustrates how interfacing according to an embodiment of the invention may provide for traversing a coupled interface, and for remote machine control, data processing and enhancement;

FIG. 11b illustrates how interfacing according to an embodiment of the invention may provide for manipulating a coupled machine or data corresponding thereto;

FIG. 11c illustrates how interfacing according to an embodiment of the invention may provide for non-speech information input and processing as commands or data;

FIG. 12a is a flow diagram illustrating a use of specificities, automatic window switching, overlaying, interface modification, data or command input/processing and enhancement, according to an embodiment of the invention;

FIG. 12b is a flow diagram illustrating automatic determining and selective inhibiting, linking, directing or processing of command/data or types thereof, a forms context and a highlighting enhancement operable by one or more local/remote users using one or more local/remote machines via a corresponding voice or other gesturing interface, according to an embodiment of the invention;

FIG. 12c is a flow diagram illustrating examples of intra-application remote grouping and intermittent context, according to an embodiment of the invention;

FIG. 12d is a flow diagram illustrating automatic switching, data locating, carrying, processing, entry, extraction, mixed context and enhancement, according to an embodiment of the invention;

FIG. 12e-1 is a flow diagram illustrating simple and complex paging according to an embodiment of the invention;

FIG. 12e-2 is a flow diagram illustrating a more specific example of paging of FIG. 12e-1 in which determined containers include message folders that may be user definable and items include messages contained within the message folders;

FIG. 12f is a flow diagram illustrating grouping, random selection, command linking and enhancement according to an embodiment of the invention;

FIG. 12g is a flow diagram illustrating command linking, data entry, imposed consistency and enhancement according to an embodiment of the invention;

FIG. 12h is a flow diagram illustrating grouping, random selection, intermittent command, automatic response and enhancement according to an embodiment of the invention;

FIG. 13a is a flow diagram illustrating virtual or actual interface modification, item referencing, separator insertion, random selection; intermittent context and imposed consistency according to an embodiment of the invention;

FIG. 13b is a flow diagram illustrating intermittent and transitional contexts according to an embodiment of the invention;

FIG. 13c is a flow diagram illustrating overlapping, addition of virtual controls, (intra application) remote control and enhancement according to an embodiment of the invention;

FIG. 14 is a flow diagram illustrating window cycling and imposed data merging according to an embodiment of the invention

FIG. 15a is a flow diagram illustrating perceived rhythmic consistency according to an embodiment of the invention;

FIG. 15b is a flow diagram illustrating intra-application and extra device coordination and information utilization according to an embodiment of the invention;

FIG. 15c is a flow diagram illustrating extra-machine grouping, coordination and coordinated enhancement according to an embodiment of the invention;

FIG. 15d is a flow diagram illustrating examples of privacy implementations according to an embodiment of the invention;

FIG. 15e is a flow diagram illustrating an interaction between two or more users, groups or workgroups (users) in which the users and/or machines may be locally or remotely located and one or more of the machines may be utilized by one or more of the

users in a direct, indirect or composited manner, according to an embodiment of the invention;

FIG. 15f illustrates initial or ongoing identification and association of different or particular users according to input device, biometrics, proximity, location, movement or other mechanism(s), according to an embodiment of the invention;

FIG. 15g is a flow diagram illustrating trapping or other processing of a mis-recitation, according to an embodiment of the invention;

FIG. 16 is a flowchart illustrating a method of conducting voice interfacing according to an embodiment of the invention; and

FIG. 17 is a flowchart illustrating a method for conducting user identification, association and interaction interpretation, according to an embodiment of the invention."

The paragraph beginning at **page 10, line 23** has been amended as follows:

"OEVI and other interfacing aspects are, however, capable of rendering particular applications, machines, platforms, speech tools/engines, included underlying interfaces or operations and other factors mere combinable implementation options. Underlying interfacing elements can, for example, include a graphical user interface ("GUI") or more sophisticated elements (e.g. 3-D audio/visual, animation, shared/remote access, and so on). User-control devices can include existing/future processes or devices (e.g. display, pointer, keyboard, tactile, biometrics, and so on), and graphic or virtual/augmented reality or other supportable environments ~~supportable~~. Interface-able devices and/or processes or "machines" can include set-top boxes, palmtops, personal digital assistants ("~~)~~PDAs"), personal information managers ("PIMs"), media production/presentation, smart appliances/phones, e-commerce, applets, servlets, add-ins, or objects, among substantially any others."

The paragraph beginning at **page 14, line 28** has been amended as follows:

"Machines 102 can also each comprise one or more still further machines at various levels, such as GUI program window segments, feature sets, question/answer "frames", sub-devices/ processes, systems, components~~,~~ or downloadables, among others.

(Conventional or non-conventional mechanisms within machines 102 that provide machine operabilities or enhancements thereof are summarily represented by operation engine 127b of machine 127.) Output produced during machine operation can further be received/ processed by output processor 103, I/O devices 105 or other suitable elements for presentation to a user, other machines or interface processor 101 elements. Machine output can also be provided directly by a machine (e.g. as with controlled audio components, a mobile phone or PDA); however, such output -as with input- might also be modified, re-directed, additionally directed, interpreted, otherwise utilized or some combination."

The paragraph beginning at **page 16, line 18** has been amended as follows:

"Server 125 is summarily depicted to indicate one or more suitable devices configurable, for example, in a so-called client-server ~~or~~, peer-to-peer type or other operation, and can include an Internet service provider or "ISP" functionality, server/firewall, corporate/home server or any other suitable server or associated operability. Server 125 can also provide or facilitate system 100 element operabilities or include any suitable elements for re-communicating/passing data or otherwise negotiating network, device, user or elemental security or other information that might be utilized (e.g. firewalls, keys, certificates, synchronization, code/data hosting, forwarding or other handling, etc.)."

The paragraph beginning at **page 18, line 5** has been amended as follows:

"System 100, however, also enables more capable implementations. For example, using more advanced recognition, interpretation or other elements, non-speech input or machine information can be stored or processed by command interpreter 115 in conjunction with other I/O, operational, "history," speech/non-speech input or other information, or further in accordance with knowledge base metrics, rules or artificial intelligence to more accurately predict, determine or respond to user intent/objectives. User interface/machine use tendencies can, for example, be monitored, analyzed and corresponding information stored on an ongoing, or end-of-session or other basis in storage 116 or one or more other local/remote storage media;. Interpreter 115 or other system 100 elements can then retrieve/receive all or portions of such "history" or analyzed history information at interface

startup or thereafter, as applicable to current interfacing (e.g. where a machine portion is operable, more specifically accessed or will likely be accessed, at one or more timed intervals, upon some event(s), and so on)."

The paragraph beginning at **page 19, line 17** has been amended as follows:

"System 200 elements also include a computer readable storage media reader 205 coupled to a computer readable storage medium 206, such as a storage/memory device or hard or removable storage/memory media; examples are further indicated separately as storage device 208 and memory 209, which can include hard disk variants, floppy/compact disk variants, digital versatile disk ("DVD") variants, smart cards, read only memory, random access memory, cache memory or others, in accordance with a particular application (e.g. see storage 116 of FIG. 1). One or more suitable communication devices 207 can also be included, such as a modem, DSL, infrared, etc. for providing inter-device communication directly or via suitable private or public networks, such as the Internet (e.g. see I/O control 105 of FIG. 1). Working memory ~~209~~ 210 is further indicated as including an operating system ("OS") ~~291~~ 211, interfacing system component(s) 212 and other programs ~~292~~213, such as application programs, mobile code, data, or other information for implementing system 100 ~~elements~~components, which might be stored or loaded therein during use."

The paragraph beginning at **page 20, line 6** has been amended as follows:

"Portions of system 100 (FIG. 1) can also be implemented as one or more low-level processes linkable to or forming part of a system 200 or other suitable operating system ("OS") or OS-like process. Such an implementation, as with conventional PC controllers, might thus benefit from reducible delays and system-wide availability, among other benefits. These or other benefits will be more readily apparent with regard to I/O controls 114 and 132 (which can form portions of a conventional BIOS or more advanced I/O controls, such as those given above), as well as with recognition engine 112, synthesis engine 131 and control portions of I/O devices 105, which tend to perform largely repetitive tasks that can also require more substantial system resources. (Any OS or programming languages capable of operating in accordance with the teachings herein can be utilized)."

The paragraph beginning at **page 32, line 6** has been amended as follows:

"A resulting conversant command set or group does not, however, require a free-form spoken-conversation approach, but can also provide other verbal or non-verbal gesturing/events or approaches (e.g. "question-answer", "lecturer-audience," "manager-assistant," supported, conversation, others or combinations thereof). Such approaches can further be implemented as automatically (e.g. programmatically) adaptable, user-selectable or some combination, in accordance with user, devices, contexts, events, gestures, portions thereof, and so on."

The paragraph beginning at **page 21, line 25** has been amended as follows:

"In summary, a "scenario" can be broadly viewed as a paradigm or perspective that can be imposed, presented or explained to a user (in conjunction with a correspondingly operable system) to facilitate conversant command use, handling (or "processing") or guiding of user expectation. "Conversant context" includes use-directed circumstances within which a user objective is to be accomplished by initiating commands or for establishing or modifying the impact thereof on user progress through successive user objectives. "Tasks" and "goals" include objectives that a user might want to effectuate or results that a user might want to produce. Tasks or goals can also be viewed as abstracted or conversantly "distilled" machine uses or purposes. "Conversant factors" include refinements applicable to commands, machines, approaches, environments, and so on for increasing interface intuitiveness or conversant user experience;, e.g. manipulating one or more command or underlying interface elements to render commands more intuitive, guiding, or otherwise more conversantly "recitable" via speech, other gesturing, events and so on, or some combination."

The paragraph beginning at **page 22, line 14** has been amended as follows:

"Consider, for example, the OEVI-like assistant scenario embodiment of FIG. 3a in which a user can move about while instructing an "assistant" or using various controllers himself; as desirable, the user can also virtually place one hand on his assistant's shoulder and point (virtually or actually) by or while reciting a corresponding objective. FIG. 3a shows, for example, how user 301a may use a head/body location or position 311a, gaze 311b, pointing mechanism 311c or change in location/position, gaze or pointing 311d to indicate a home or

business, office 313 and computer/display 312a (or one or more other real or virtual location or machine portions), or further, to indicate one or more presented objects 312a1 presented by display 312a of computer 312. Such indication(s) may, for example, cause a corresponding recited objective to correspondingly affect one or more of the illustrated or other machine portions. (Location, status, multiple user, group, local/remote handling and other enabled aspects are also discussed below.) ~~The a~~Assistant 301b can comply with simpler recited user objectives; the assistant can also switch, distribute among or link machines or commands, imply specificities/designations, perform current/later command enhancement, and so on, thus enabling minimal user recitation to produce even complex, multiple local/remote machine effectuated objectives. An assistant can also guide an inattentive or physically, emotionally or technologically challenged user, for example, by receiving and resolving partial recitations, providing suggestive feedback, advancing user progress, preparing for further objectives, and so on."

The paragraph beginning at **page 24, line 24** has been amended as follows:

"A grouping can also include separately presented control/data elements that are predicted or observed (e.g. at runtime) to be mentally organize-able, similarly referable or otherwise usable by a user as a group or portions of that group (e.g. 321c-e, 321b and 321f, 302-4 and 306, patent claim portions with or without graphic, textural or other references/modifications thereto, clauses, notes, annotation types/sources or combinations; multimedia portions, control types, machines or machine types 306, 307a-c where or through which information might be located or which might be operated, and so on). See also Designations below. (While referencing a group as a whole is useful, the ability to similarly perceive/reference one or more group elements of the same or even differing groupings is found to facilitate generic applicability, flexibility and intuitiveness.)"

The paragraph beginning at **page 25, line 3** has been amended as follows:

"An OEVI-like "extended grouping" context further includes one or more groupable items associate-able by a user (and an OEVI-like conversance-enabled interface) with the group designations and a corresponding toolset. Examples include: email folders, emails and tools; file folders, files and tools; available windows, a target window and window controls for that

window; home theater or multimedia production/presentation device sets, operational modes and controls; form segments, fields and tools; instruments/orchestrations, instrumentations/effects or other attributes, and controls; home, office, car or user reference-able smart devices, like specifics and controls; other machines, presentations/modes and controls, and so on. (For example, FIG. 3b extended groupings include: folders 322a, emails 321a, controls 324a-b and related windowing of 302b and 321-2; window portions 302-304, 321-323 and 331-332, an included designated portion, such as message 321b or messages 321e, and windowing tools summarily depicted as 302b; 361a-b or 307a-c, 361b or 307 portions and 364 or 307 controls, among others.)"

The paragraph beginning at **page 26, line 26** has been amended as follows:

"In many cases (during creation), context/sub-context accommodating commands might already exist due to prior application of context or other conversant aspects, and sub-context might provide more of a creation (or other handling) check to assure coverage of expectable commands according to context or other conversant aspects as well. In other cases, application of sub-context enables contextual or other conversance holes to be identified and filled as needed. In the OEVI, command permutations are provided for accommodating ongoing, transitional and sub-contexts (e.g. see the ~~FIG. ____~~partial command chart in the attached appendix); as noted above, these can also be detected and accommodated at runtime via suitable analysis (e.g. of history information) and initiating of corresponding operations."

The paragraph beginning at **page 27, line 4** has been amended as follows:

Tying command recitation or handling to more generally applicable conversant context (or sub-contexts) is found to facilitate, for a user, flexible semantic continuity and intuitive recitation, and implementationally more accurate and efficient command handling. For example, users tend to intuitively utilize increasingly consistent recitations, enabling fewer commands to be provided and awkward recitation avoided. Users also tend to recite commands more confidently where conventional conflicts between control and data handling and other recognition inaccuracies can be avoided; common user mis-recitations can also be better identified and thus trapped or otherwise handled, and unnecessarily user recited machine operation steps can be identified and corresponding enhancements implemented, among other

useful results. See, for example, FIG. 15g. (Conversance enables handling that can not only determine, accommodate and assist, but can also modify or otherwise "guide" user perspective or expectation.)

The paragraph beginning at **page 28, line 13** has been amended as follows:

"A user is also found to largely subliminally use even differing coexisting contexts as needed, e.g. utilizing data search fields differently than also presented (resulting) data lists, despite the confusion or inexactitude that might at first be expected to exist given prior (e.g., PC program) description or use~~; e.g.~~, for example, as with presentation contexts 331a and 332a of FIG. 3b. Underlying interfaces also provide familiar elements/operations, the modification of which can further be used to guide a user with regard to other, even unrelated or otherwise unavailable machine utilization~~; e.g.~~, for example, other programs, using a PC, a cellular phone or stereo similarly, or further in conjunction with one another, and so on. (Note that creation is affected since automatic, modifiable automatic or explicit user-directed "switching" between coexisting contexts may well need to be identified and accommodated by adding/modifying commands or enhancements to facilitate switching among coexisting as well as other contexts. In many cases, identifying a default entry point to the coexisting context combination and enabling "exiting" commands that are shared by the combination is sufficient; in other cases, location or other context based commands/permutations might also be used to provide case-specific entry/exit.)"

The paragraph beginning at **page 30, line 11** has been amended as follows:

"Consistent with the OEVI and emailing, for example, "simple ongoing tasks" might include flagging one or more displayed or not displayed emails/folders, deleting them, assigning them user priority organizations, other characteristics, otherwise moving through an email list, or other objectives. "Related ongoing tasks" might include having an assistant read back emails, headers, locations or other supporting/descriptive information (via graphics, speech, etc.). "Related intermittent tasks" might include: finding emails to/from someone, on a particular subject, sent/received within a time period, having some priority range or some combination; responding by fax or phone; scheduling; alerting a secretary or others; taking

notes; distributing information, etc. "Lesser-related intermittent" or "new-application" tasks might include reviewing/modifying a flowchart, program code, a presentation, or other information –perhaps relating to an email or copyable information therein. Lesser related intermittent tasks may also include, for example, temporarily interrupting using computer 312 of FIG. 3a to control light 313c1, a television, stereo/gaming equipment, and so on. "More complex tasks" might include alerting others as to scheduling, manipulating a presentation with or without "opening" it, sending one or more specifiable phone, mail or email responses – perhaps according to a sender, source, and so on."

The **2 sequential** paragraphs beginning at **page 30, line 25** have been amended as follows:

"(The OEVI implementation, being subject to existing tools, is particularly limited with regard to guiding/facilitating lesser-related tasks; however, it makes extensive use of implicitly known/anticipated information. The OEVI provides for such interfacing as responding to a command by determining and utilizing relevant current task information in a successive command. For example, reciting "Find more messages" causes the interface to determine that sender information is needed, find the information, initiate the OE find option, enter the information and initiate a search. See, for example, FIG. 12d. "Call sender" or recipient, "Call <X> at home", and so on further cause the OEVI to lookup a corresponding target (e.g. phone number of the sender/recipient) using a corresponding machine (e.g. one or more local or remote personal, group or public address books) and implement a resulting resolved objective (e.g. place a call or display correspondence information.)

However, more subtle inferences based on runtime determinable user intent/objectives that might even conflict with predetermined courses of action are not supported in the particular OEVI embodiment; for example, use of particular address books or responses in particular circumstances might be more efficiently/effectively determined and implemented at runtime. (See, for example, "history" above.) ~~A partial list of OEVI commands is given in FIG. ____.~~)"

The **2 sequential** paragraphs beginning at **page 31, line 22** have been amended as follows:

"For example, conversant commands enable a PC user to perform operations "outside" a displayed pop-up window, window segment, program or PC, switch machines, use them in conjunction with one another, work with different users, user groups, moderated groups, and so

on. PCs and different, even unrelated machine portions can also utilize a similar "language" or similar/convertible implementation (that can further be extensible as might be applicable). See, for example, FIGS. 10b and 15a-c.

A different handling decision nevertheless exists as to whether enabling the user to use differing expression itself creates ambiguity of command use or otherwise conflicts with conversant aspects, e.g. where one window portion includes scrolling and another window portion used in conjunction therewith does not include scrolling, and the difference, rather than facilitating recitation, might create confusion.) Again, the particular OEVI implementation utilized required such analysis and determination to be conducted entirely as part of command creation."

The paragraph beginning at **page 32, line 4** has been amended as follows:

"Also consider utilization of OEVI-like conversational factors. Conversational factors enable intra or inter command structure/content to be created in a manner that facilitates user progress. Such factors include, for example, smooth verbal, physical or other command portion recitation/enunciation, and "rhythmic flow" (i.e. establishing or propagating an expectable ongoing, transitional or new perceived meter, pacing or other command/command element flow). A useful user recitation analogy might, in retrospect, be a music sight reader using immediate or ongoing musical rhythm, meter or phrasing in a predictable manner as a guide as to "what to play" next; as a result, continued sight reading becomes an increasingly simpler and more automatic "user default". See, for example, FIG. 15a."

The **2 sequential** paragraphs beginning at **page 32, line 29** have been amended as follows:

"Conversational factors also include imposing "balance" (which attempts to assure that tasks supportive and complimentary to one task are also available and operable), and imposing "consistency", which attempts to assure that a task supported in one instance is similarly supported and (at least apparently) similarly operable in other expectable instances, among other examples. See, for example, FIGS. 12g and 13a-b. (Thus, conversational factors can be used in creating recitable commands or to causing actual/apparent underlying machine operation, often via recitation additions, or enhancements.)

As with other conversant aspects, conversational factors are found to be co-supportive, such that balance and consistency are achievable, for example, using the aforementioned context, tasks/goals, linking, enhancements, feedback or command/data input; a combining of factors has further been observed as tending to provide further improvement.) In the OEVI, for example, the above-noted "Find more <optional specificity> messages <optional specificities>" command is also available in co-supportive forms of "continued commands" within the OE Find window, enabling consistent and rhythmically flowing continuation of finding particular messages (e.g. "Find more *flagged* messages". Abbreviated co-supported forms are also provided as supportive of potential changing flow (and also in support of a changing context in which "more" might no longer apply, and in providing only needed information to an assistant). Within the Find window, for example, it is also sufficient to recite "*Flagged* messages"; "*All* folders" or messages; "Messages *in <X> folder*", "To sender", "Regarding <X>", and so on. (Multiple or more inclusive explicit or implied specificities or conjunctives, such as a leading "And...," can also be used in such cases to provide combined or more refined/complex referencing variations, in accordance with adding to versus modifying currently established conditions/objectives.)"

The **2 sequential** paragraphs beginning at **page 34, line 21** have been amended as follows:

"Turning again to FIG. 3b, assume that a user has recited a command to launch OE (e.g., "Run email", "Receive email", and so on) such commands are unlike the OEVI "Send an email...", which also brings up a new-email form, further fills, facilitates addressing or even sends a new-email.~~; the~~ The OE window is, in the OEVI case, displayed and set to point to some extended grouping element (e.g., email 321b in Inbox 322b). (Note that an OEVI "Receive email" command also checks an email server and downloads any new messages. It was also decided for the OEVI that, given limited control information, re-instantiation of new program instances should be avoided as often not available and otherwise causing an inconsistent interface response; thus, an already running target program is rendered "current" as needed, while a not running one is first instantiated.)

The OEVI assistant scenario imposes largely explicative commands that "tell the assistant to perform a task" and implying not otherwise stated specificities. Thus, for example, a user can tell his assistant to handle listed messages (e.g. "Read these next 3"; "Flag last 3", "Delete

last 3"). Rhythmic flow is also implemented in accordance user tendency to mentally combine certain recitation permutations (e.g. "these next") or multiple beats (e.g. "Delete), which are also applicable to movements, biometrics or other recitations, and so on. Added enhancements further adjust pointing, commands, data, and so on, or provide feedback, so that a user can confidently expect particular results and can quickly rattle off ongoing use-objectives (see below)."

The **4 sequential** paragraphs beginning at **page 35, line 12** and have been amended as follows:

"The OEVI further similarly implements recitable specificities as to other grouping, extended grouping or other contexts. Thus, a user can also tell his assistant, for example, to "Scroll *folders*...", causing the interface to execute a corresponding ongoing or intermittent context (by switching to folders if needed, scrolling the folder list and returning to one or more elements if needed -in this example- to the same email **321b**); consistent "Scroll messages" or "Scroll emails" (using or not using distillation) are also provided in accordance with conversant factors. See also FIG. 12c. A user can also similarly effectuate operabilities with regard to other presented elements (e.g. within a current window, virtual/augmented reality scene, frame, and so on (e.g. "Scroll left 3") or a sufficiently specified current or not current machine portion (e.g. "Scroll <program/ window name or type>...").

Grouping and extended grouping contexts facilitate not only reciting objectives relating to a "current" or "new target," but also using targets in conjunction with one another (e.g. moving/copying items between or among containers, such as folders or graphic groups; affecting items within even not-current containers; affecting container portions intermittently, transitionally or automatically.) A user can, for example, "Delete next n folders", open one or more folders, modify the names of folders, and so on intermittently, or create new folders as a new goal (using invoked OE features). A more advanced implementation would, however, also enable creation to be effectuated semi-intermittently, for example, by returning to a starting or other pre-command condition once the creation has been completed, in accordance with more information or other runtime determinations.

(Note that NS and other recognition engines require explicit word matching, with options being specified within a command list. In such cases, particularly user-modifiable designations

(e.g. email folder, contact, current message, file folder merge attributes, names, titles, and so on) will need to be entered and integrated with commands during creation, execution, conversion or other handling (e.g. by polling machine information, downloading or other suitable techniques). For example, NS can be provided with the entirety of "Move next 3 to *Business* folder" by updating a "folder type" command-list within a programmed command that includes "Move..." with Business and other folder designations. See, for example, FIG. 12h. Depending on the particular implementation, such modification (or attaching, such as in the below Command examples) can be conducted using local or remote information during creation, at the startup of runtime, responsively to an otherwise recognized command, and so on, or some combination.

While more difficult to satisfy conversant factors, more extensive specificities can also be similarly created, recited or otherwise handled. Note, however, that not all objectives can or *should* be so neatly provided in light of other conversant aspects that might be used. For example, "Goto page 3" sufficiently tells an assistant (as, for example, shown in FIG. 3b) what to do, since of the illustrated machines, only word processor 116 *has* pages.~~, but might not~~ Such a determination may not, however, be sufficient in other multiple-machine circumstances. Continuity and other conversant factors would therefore suggest an addition/replacement, such as: "Go" or "Goto" (and the like) up/down, backward/forward, page, section, footnotes etc. "*in <program or window name*"; "*<program or window name> go...*"; and so on. (As noted, conversant aspects attempt -other factors being equal- to exclude recitations that create a user expectancy in one instance that will be confusingly responded to in one or more other instances.)"

The paragraph beginning **at page 37, line 8** has been amended as follows:

"OEVI intermittent versus transitional contexts were typically determined as follows. Both typically occur at machine operation (e.g. window portion-to-window portion) transitions; these can be identified via command-creating user interaction, programmatically invoking features and monitoring resulting machine responses (e.g. for windowing events), prior knowledge, conversant interface communication with a machine configured for providing such information, and so on. If a result completes a determined goal and does not require a presented portion that replaces a current one, then an intermittent context has typically been identified. If, instead, the ~~goal~~result does not complete a goal or such replacement occurs and it is infeasible

or undesirable to avoid such replacement, then a transitional context has typically been identified. (Other instances might also be desirably implemented, or some combination.)"

The paragraph beginning **at page 38, line 4** has been amended as follows:

"As an example of form contexts, the OEVI transitional "Add new contact" might be recited, causing the interface to invoke window 306 (FIG. 3c), switch to the "Name" window portion, highlight and select "First". ~~("Modify <home/office> address, or "Modify <X's>…", etc.~~ Thereafter, in accordance with a location context, a name or other "form sequence" can be recited with or without recitation of controls. For example, enhancements can automatically advance responsively to the separately spoken "John" "S." and "Smith," or in a more advanced implementation, "John S. Smith" can, for example, further be parsed for name versus letter or individual word recitations (e.g. for "John Simon Smith"). A user can also reference fields directly in abbreviated form or with greater specificity by reciting, for example, "First name" "John", "Middle initial S." or "Middle name "Sam", "Last name" "Smith", among other examples. (In the OEVI, "initial" and "name" respectively provide for entering capital letter plus period, or a name in conjunction with NS, which is literal and ignores such distinctions.) Note, however, that mere recitation of "First", "Last" and "Middle" were determined to be confusing and contrary to rhythmic flow as compared with other recitations that might also be used (being single words), and were excluded from the particular OEVI implementation. ("Modify <home/office> address, or "Modify <X's>…", etc. may also cause a window portion or sequence corresponding thereto to be invoked or highlighting or other feedback to be provided, e.g., as shown in FIG. 12b.)"

The paragraph beginning **at page 39, line 18** has been amended as follows:

"The FIG. 4a flow diagram illustrates a conversant command creator 111 of FIG. 1 according to an embodiment of the invention. In this example, command creator 111 is implemented as conversance-enabled analyzers operating in conjunction with user/machine directed knowledge bases including the above-noted conversant aspects. Command creator 111 of FIG. 4a includes application analyzer 401, conversant user-interface analyzer 401a and operability analyzer 401b, each of which is capable of determining command aspects in a separate or combined, and typically iterative manner. Application analyzer 401 further includes

application-analysis engine 411 and application knowledge base 412. User-interface analyzer 401a further includes conversant I/O analyzer 402, I/O knowledge base 403, conversational factor analyzer 404, iteration analyzer 405, genericizer 421 and approach analyzer 422. Operability analyzer 401b further includes operation analyzer 406, operation knowledge base 407 and optimizer 408."

The paragraph beginning **at page 41, line 27** has been amended as follows:

"Application analysis engine **411** receives machine information of a target machine and determines applicable application/purpose categories ~~of~~ corresponding thereto. Such determination can be conducted via user interaction or more automatically (e.g. according to category selection rules/metrics); however, given currently limited machine classifications, either or both should provide for extensibility at least at present. More conversant target machines (or storage/re-communication devices such as servers) might, for example, provide categorization information directly or might provide machine characteristics from which such categories might be more directly determined or verified, for example, according to a conversance standard. (For example, a raw or weighted comparison can be conducted with existing uses stored in application knowledge base **412** and a sufficiently close existing use or new one can be assigned. Communication of such information can further be conducted via downloading, polling or other suitable mechanisms or methods."

The **2 sequential** paragraphs beginning at **page 47, line 10** have been amended as follows:

"Note that I/O, operational or other "known" aspects need not be wholly contained within knowledge base elements, and external or remote information can also be used. For example, a user's existing interfacing information (e.g. grammar, vocabulary, speech files, preferences, data, and so on) or existing machine or speech engine information can be provided via mobile code, such as applets, loading or push/pull retrieval, among other polling/downloading alternatives (e.g. see, for example, above). Other iterative or non-iterative processing ordering or other alternatives might also be applicable to a particular implementation. (See, for example, FIGS. 1, 9b, 9e, 10b and 15b.)

FIG. ~~3b~~4b illustrates an I/O analyzer implementation in greater detail. As shown, I/O analyzer 402 includes environment engine 402a for determining presentation elements of one or

more machines, and conversant element engines (e.g. objectives engine 402b for determining user objectives, context and task engines 402c through 402d ~~for determining conversant contexts and tasks respectively~~ for determining conversant criteria according to which commands are to be formed. I/O analyzer 402a also includes operability engine 402e for determining applicable operabilities in conjunction with machine-capability analyzer 301b."

The paragraph beginning **at page 48, line 5** has been amended as follows:

"Moving to the lower portion of FIG. 4a, machine-capability analyzer 401b provides for analyzing underlying machine characteristics, including controls, capabilities or feedback, determining whether one or more potential user-commands are implementable and, if so, determines code for implementing such user-commands. Analyzer 401b can also determine whether more than one control might be used (differing results thus far being determinable via user interaction) in accordance with user input, machine communication or knowledge base 307 rules, metrics or data (e.g. as with application or I/O analyzers)."

The **2 consecutive** paragraphs beginning at **page 49, line 1** have been amended as follows:

"Operation KB 407 includes existing command KB 407a, underlying machine KB 407b ~~and~~ speech engine KB 407c and user/group KB 407d. Existing command KB 407a and underlying machine KB 407b include elements for enabling communication and operability command execution with known or new machines. Existing command KB 407b includes more generally applicable and communication information, such as pervasive commands 471, e.g. OS commands, general functions (see examples below), common commands (such as PC save, load and edit commands), and so on. Also included are connection and OS information 472, command/machine distribution information ~~472~~473 (e.g. associated command/data structures, memory/storage locations, and so on), communication and command delivery or other communication protocols ~~473~~474, and other information ~~474~~475. Underlying machine KB 407b includes machine capabilities ~~475~~476, available controls ~~476~~477, available parameters ~~477~~478, memory map information ~~478~~479 (e.g. data/attribute or control layout of programs/devices), enhancement information ~~479~~480 (e.g. desirable enhancements and implementation), and linking information ~~480~~481 (e.g. for below discussed overlaying or other

multi-machine capabilities). Other machine interfacing information ~~481~~482 can also be utilized.

Speech engine KB 407c includes information pertaining to communicating with a particular speech recognition or synthesis engine, such as available control mechanisms 491 (e.g. information handling, synthesis speed/pitch, and so on), command parameters 492 (e.g. for other available controls or control variations), form information 493 and other information 494. (Note that attributes of other event/gesture processing engine peculiarities can also be utilized, or one or more front or back end knowledge bases can also be utilized and other information can be utilized in accordance with a particular application.)"

The **2 consecutive** paragraphs beginning at **page 50, line 18** have been amended as follows:

"Operability can further be provided as more globally applicable or with respect to more particular program(s), device(s), application(s), approach(s), environment(s) or types or portions thereof (e.g. implicit/explicit video conferencing audio, video or graphic tracking; adding, modifying or indicating multimedia presentation elements; creating, modifying or transferring email, fax or other documentation\communication; operating a home entertainment system, etc.) Such operability can also be applied to control, data entry, feedback or some combination, as might be suitably executed by one or more machines. The use of conversance also enables commands to apply similarly for operabilities that, in a conventional sense, would otherwise be considered data input or other data handling. For example, continued commands enable the selection of multiple disjunctly positioned data items (e.g. "Select first 3" "And select last 3 <item type>"). See, for example, FIG. 12g.

FIG. 4c further indicates how, in addition to storing resultant command portions in one or more knowledge bases or some other storage medium (e.g. see FIG. 1), commands can also be distributed to other machines. Command distributor 495 provides not only for distributing complete commands to independently operable or otherwise remote systems, but also for distributing command portions. Thus, for example, recited commands might be interpreted locally and executed by a remote machine, interpreted remotely and executed locally (e.g. where a current machine lacks sufficient processing capability for interpretation) or interpreted and executed remotely. (As noted earlier, distribution might also be conducted in conjunction with a suitable command converter; this can avoid duplication of machine information where a

separate knowledge base or other storage medium might be utilized for storing machine information.) See also FIGS. 9e and 10b."

The paragraph beginning **at page 52, line 15** has been amended as follows:

"Command groups can also be utilized during execution, conversion or other handling. In the OEVI, for example, predetermined commands not only provide intuitive, easily enunciated command permutations in reciting successive commands, but also using different OE windows or other programs for similar purposes (e.g. manipulating new or found emails and addressing them, manipulating file folders generally or word processing document portions, and the like). More advanced runtime processing can during execution or conversion can also facilitate a consistent user experience (with or without explicitly flagging correspondences during creation or thereafter) via the above noted monitoring or other suitable mechanisms. (Conversion can, for example, modify commands in accordance with language, dialect, cultural or other relationships to more suitably affect command structure/verbiage, grouping or other applicable conversant relationships.)"

The paragraph beginning **at page 54, line 3** has been amended as follows:

"A successive command can, for example, include the same or different ones of the afore-mentioned portions, as with example 1.~~; while~~ While typically conversantly related to a prior command 501a, such a successive command can be otherwise independently interpretable/ executable or otherwise handled. Such a successive command can also be perceptually linked (though not explicitly interpretively linked) to other commands, which linking can be supported or otherwise guided by conversant aspects, such as in the above examples. (For example, successive claim dependencies can be added to a patent claim chart, much like the above email modifications, successively using the command "Next [patent claim] depends on <N>" or "Next is <a or an> [independent] <method or apparatus> claim". Note that the bracketed portions might be optionally recitable as more specific designations, such as after pausing or for causing operation to shift to claim charting from some other objective type, goal or presentation, such as was discussed with reference to FIG. 3c.)"

The paragraph beginning **at page 55, line 1** has been amended as follows:

"Thus, for example, a single user execution system implementation can determine/utilize particular local or remote user information, or "track" and respond to different user objectives or elements (e.g. actions, targets or other specificities) thereof. A multi-user execution system implementation can further determine/utilize local or remote user/group information, "track" and respond to elements of one or more local/remote user recitations, or facilitate local or remote individual, workgroup or moderated workgroup interaction (e.g. by member, status or member-groupings). See, for example, FIGS. 9a through 10d."

The paragraph beginning **at page 55, line 21** has been amended as follows:

"Turning to FIG. 5b, an OEVI base command (e.g. "Send an email", "Fax a <correspondence>" or "Phone <name>") typically corresponds with the simplest form of a task or objective (e.g. "Send an email"), which often includes an action as a root (e.g. "Send"), a connecting portion (e.g. a, an, the, that, these, and so on) and an object of that action as a designation (e.g. "email"). When corresponding to the simplest form of a task, the root command typically excludes modifiers (which typically provide a modifying attribute of a corresponding designation and can affect execution of a corresponding enhancement). A base command can, however, also include implied other command portions (e.g. specificities, extensions, and so on). For example, a particular base command might correspond with an often used command permutation, a command permutation in accordance with a user preference, security or other attributes, and so on. In such cases, a base command might provide a simpler way of reciting a more complex command, or serve other purposes in accordance with a particular application."

The paragraph beginning **at page 56, line 19** has been amended as follows:

"Extensions (e.g. extension 502) are typically coupled to a root command and can provide additional explicit task(s) (e.g. permutations of a base command), and enable improved underlying machine operability or facilitate continuity, simplification, expectancy or other interface utilization (e.g. via greater specificity). A user, for example, typically adds an extension to a base command during recitation for defining alternative tasks corresponding to the same, or further, other base-commands (e.g. "to [person] C", to C "at work", "to C1 and C2 at the Z office", "using WordPerfect" or "using [MS] Word", "email D to B ", and so on).

Extensions or alternative base commands can also be used to provide "reversible" commands, the user perception of which is of freely mixing command recitation of command verbiage (e.g. "send D to B using OE" versus "OE send D to B", "make the 4th [displayed] column Z" versus "make Z the 4th column", "email the X files to A using [my] <machine-name>" versus "<machine-name>... send the X files to A", and so on). (Note that such a system of commands is adaptable to the above-noted placeholder, mapped or other suitable applications of distinguishable command elements to corresponding commands.)"

The paragraph beginning **at page 57, line 2** has been amended as follows:

"Extensions can also be used to extend a base command recitation and can provide a more explicit statement of the base command. For example, the base command "Find more messages", while meeting conversant factors on its own, is found to be rhythmically inconsistent with "Find more messages *to sender*"; the re-stating extension of [~~send~~find more messages] "from sender", which is implicit to or "presumed by" the base command, is therefore also provided. (In such a case, rhythmic flow also provides a command creation metric determining the impact of command length.)"

The paragraph beginning **at page 57, line 23** has been amended as follows:

"Extensions also tend to be perceptually different from other command portions. Due to limitations of current speech recognition, it is sometimes necessary to distinguish commands from dictation by omitting words from commands (e.g. pronouns). It is found, however, that interface intuitiveness can actually be improved by limiting (most often) such "incomplete recitation" to base commands and limiting the number of base commands. (A user is found to actually use a conversant interface more intuitively where a more limited number of incomplete base commands is combinable with a greater number of "completely recitable" extensions. During execution, incompletely stated base-commands also provide a more implementationally and recitably consistent basis for distinguishing commands and data with or without extension. (A user can also apparently be guided in this manner toward reciting resulting "shorter" and more easily recited base-commands in succession.)"

The paragraph beginning **at page 58, line 5** has been amended as follows:

"Enhancements 503 can provide peripheral capabilities that can further be related to other command portions. For example, an enhancement of "Send an email to C" might identify B as a current user, activate B's PC, PIM, etc. (e.g. if remotely initiated), run an email-handling program or applet (if various programs might be run), initiate a new email message, address the message, and prepare (e.g. via curser repositioning) for an email message subject. Thus, a next user command can simply enter the subject (with corresponding enhancements of advancing the curser to the email message body, providing for additional addressing, etc) or redirect operation to a less-often utilized successive capability. See, for example, FIGS. 3a-c, 10b, 12, 13b-c and 14."

The paragraph beginning **at page 58, line 21** has been amended as follows:

"Designations ~~203~~541 provide for tool, subject or object identification to which a command/ data can be directed. They can be used to implicitly or explicitly identify one or more users, applications, machines, tools, subjects or objects -even manners of proceeding (via base, extension, enhancement, subsequent command elements, and so on), among other aspects."

The paragraph beginning **at page 59, line 24** has been amended as follows:

"Commands 500a-c also demonstrate ~~an~~ examples of the application of conversant interfacing. The OEVI presentation context for command-set 500c is generally transitional, e.g. expectable in conjunction with moving from a review of existing/new email messages, working within other programs, etc. to creating/addressing a new correspondence. More than one correspondence can, however, also be created in succession or further manipulated ~~in a~~ sequentially or otherwise."

The paragraph beginning **at page 60, line 5** has been amended as follows:

"Base command 501a is also compatible with conversational factors. It is, for example, found to be intuitive, easily enunciated, providing of rhythmic flow and consistently implementable (e.g. using enhancements, such as given above) alone, with the included extensions, and typically with other commands that might expectedly be used in conjunction

therewith (or further commands provided to maintain conversance). Adding a detail/selection descriptor or "specificity" to subject "A", for example, is found to be mentally combined with "A" to maintain the simple triplet -and more importantly- a cohesive rhythmic "feel" or "meter" of base command 501a, permutations or in conjunction with other expectable command portions."

The paragraph beginning **at page 60, line 24** has been amended as follows:

"FIG. 6a illustrates a conversant command executer 600 that provides for initiating/ executing speech/non-speech based commands, data entry or feedback according to an embodiment of the invention. As depicted, recognition and command interpretation elements 112 and 115 are configurable such that they are capable of operating in an OEVI-like manner more consistent with prior (matched, non-conversant, individual-user and speech-only) implementations. FIG. ~~6b~~4d further illustrates how, following even prior recognition, a command interpreter can be implemented in a separate manner as a conversant "converter" prior to or following conventional command interpretation 115a or 115b, or in a more integrated manner, such as will be next discussed in greater detail. (Converter 115a, 115b can also include, for example, converter 113 (FIG. 1) or other system 100 elements. Thus, surprisingly improved accuracy and conversance can be achieved via an OEVI-like configuration (e.g. using overlaying), or further benefits can also be achieved via further reliability, conversance or other execution time processing."

The paragraph beginning **at page 61, line 14** has been amended as follows:

"FIGS. 6a and ~~6b~~4d also illustrate how further improvements are also achievable via shared-resource utilization or integration of one or more executor 600 elements within a more advanced command interpreter 115 capable of also providing recognition reliability/application support, a more advanced interpreter or both. Systems 600 and 600a, for example, (with or without further superposition) enable the utilization of "hooks" to conventional or future interpreter processing (e.g. via function calls) or the interception of commands (e.g. in a similar manner as with a superimposed implementation). Systems 600 and 600a also enable (with or without further superposition) the adaptation of existing or future interpreters for more integrated approaches that might be utilized. Thus, the present executor example (as with other

element examples discussed herein) enables various existing interpreters to be utilized with or without modification in accordance with a particular application."

The paragraph beginning **at page 62, line 1** has been amended as follows:

"Within a more typical recognition engine 112 embodiment, recognition control 601 provides for analyzing speech input, determining therefrom "spoken" word or phrase input or input "recognition alternatives", and transferring the (verified or yet unverified) results to command interpreter 115. Language analyzer 602 provides recognition control support for recognition control 601 including determining speech corresponding to received input in accordance with a predetermined vocabulary, grammar rules or further non-conversant or conversant analysis or refinement application-support aspects (e.g. see also FIG. ~~6b~~4d). Particularly recognition aspects of reliability or application/machine support can also be provided via elements 621-627~~626~~ or within a suitably advanced recognition engine providing, for example, a conversant recognition reliability engine 603 or conversant application support engine 604."

The **3 consecutive** paragraphs beginning at **page 62, line 17** have been amended as follows:

"Command/data engine 611 provides for receiving speech/non-speech input, differentiating commands and data, and processing and outputting data, commands or both. Command interpreter 115 operation is further supported by conversant command engine 612, which determines more specific command, data or feedback output in conjunction with reliability determination by reliability engine ~~622~~623 (or 613), application/machine support engine 614 or portions of other of refinement and support elements 621-626. Designation engine 615 provides for explicit/implicit subject, object and specificity determination, and enhancement engine 616 provides for interface/machine operability, information utilization and user experience enhancement.

Of the reliability and application/machine support elements 621-626 (or "support engine" 620), user-use engine 621 provides for identification of one or more users, and for resolving application, machine or command element "utilization". Security engine 622 provides for enabling determinable user access, filtering, storage/retrieval and operability with regard to

various applications, machines, machine features, enhancements, etc. Reliability engine 623~~622~~ similarly provides for further recognition or interpretation refinement, more suitably in accordance with conversant attributes such as those already discussed. Conversance engine 624~~623~~ provides for receiving and analyzing speech and non-speech elements (e.g. input or non-finally processed input) in accordance with conversant aspects, and returning recognition, command-input differentiation and command interpretation refinement information. History engine 625~~624~~ provides for analyzing history or current input alternatives in determining recognition/ interpretation in accordance with more likely user objectives or system efficiency. Parser 626 provides for determining or isolating one or more input/response portions, or further, determining one or more characteristics of the portions. Information retriever 627~~626~~ provides for communicating user, machine or other information for use in or as a result of recognition, interpretation or application facilitation. (Information retriever 627~~626~~ is further capable of communication generally with other elements, and can comprise a shared resource for other elements of system 100 of FIG. 1). A simulator, emulator or other machine 631 is further coupled to and operable in conjunction with the aforementioned executer 600 components, for example, as is discussed elsewhere herein.

FIGS. 7a-7d illustrate command interpreter 115~~611~~ elements of command executer 600 (FIG. 6a), while FIGS. 8a-8g illustrate support engine 620 elements, according to an embodiment of the invention. The depicted implementations utilize, within each of the command interpreter and support system elements, a "control" element (given first) plus one or more remaining analyzers/engines for providing supportive analyses or other operations, one or more of which might be utilized to receive, process and return information. Among other considerations, such an embodiment enables adaptability to various existing or more advanced speech engines, other control elements, analyzers/engines or other resources that might be utilized, and simplifies implementation, maintenance and updating. Other implementations might, however, also be utilized."

The **2 consecutive** paragraphs beginning at **page 63, line 23** have been amended as follows:

"Beginning with FIG. 7a, an exemplary command/data input engine 611 comprises input control 701, speech content analyzer 702~~703~~ and non-speech content analyzer ~~704~~703. Input

engine 611 provides for initiating the remaining elements as needed for recognition reliability, command/ data differentiation and command selection/utilization, or a portion thereof. Speech content analyzer 702 provides for initiating speech or other expression input e.g. speech, motion, audio, graphics/, video, etc.) determination, while non-speech content analyzer 703 provides for initiating reliability/operability determination with regard to mousing event or other largely static event input, ~~both~~and provides for receiving, transferring and analyzing respective information as needed.

Recognition reliability analysis results can be returned to input control 701 (or raw input resolution can be further returned to recognition engine 112 (FIG. 6a) in a suitably advanced embodiment) or further processed within command interpreter 115 (FIG. 6a), while command-data differentiation processing can be conducted in accordance with the following."

The **4 consecutive** paragraphs beginning at **page 64, line 10** have been amended as follows:

"FIG. 7b illustrates how an exemplary command engine 612 includes operation analyzer 711, root engine 712, linker 713, base operation engine 714, operation enhancement engine 715, data enhancement engine 716, feedback engine 717 and environment engine 718. ~~operation~~ Operation analyzer ~~611~~711 provides for receiving alternative-resolved input and utilizing other elements (e.g. reliability, history or conversance engines, etc.) for determining corresponding commands or command enhancements. Root engine 712 provides for determining an applicable root command in conjunction with received input. Linker 713 provides for determining whether later (typically successive) input forms a part of a command/data input already received and causing a linked command to be invoked accordingly (e.g. enabling "send an email to X" and "send an email" -breath- "to X" to provide similar results).

Of the remaining command engine ~~712~~612 elements, basic operation engine 714 provides for resolving input (e.g. in conjunction with designation engine below) and causing a corresponding basic operation (with any extensions) to be executed. Enhancement engines 715-717 similarly provides for determining and causing to be executed any operational, data or feedback processing enhancements corresponding to the basic operation and other applicable engine results respectively (e.g. the above-noted special commands, location or other context/sub-context, intermittent event input or other history, etc.). Environment engine 718

provides for assuring a determinable degree of interface continuity and appropriateness within a given environment or approach and for any "environment" specific processing (see above).

FIG. 7c illustrates how an exemplary application support engine 614 includes application support control 721, application-machine interfacer 722, special command engine 723, application parameter engine 724 and suppressor 725. Application support engine 614 provides for interoperation in accordance with a particular application (which can include one or more linked or interoperable machines operations, or machines themselves). Application support control 721 provides for inter or intra element initiation, instantiation or communication (e.g., operation, state or parameter passing).

Application-machine interfacer 722 provides for determining a communication mechanism appropriate to operating a particular current machine or machines (e.g. as stored or obtained via communication of such information). Special command engine 723 provides for determining unique-operability machine procedures that might be utilized (e.g. operating a display, switching programs, moving data, causing an otherwise unavailable operation to appear to be available, such as already discussed). Application parameter engine 724 provides for establishing user, interaction or user-status parameters (e.g. officiator) on an application, machine, machine function or task basis; these are generally determinable via a user, storage 110 or information retriever 627 (for remotely stored such information), and corresponding security information is determinable via security engine 622 (FIG. 6a)."

The **6 consecutive** paragraphs beginning at **page 65, line 22** have been amended as follows:

"FIG. 7d illustrates how an exemplary designation engine 615 provides for receiving designations from (typically) command engine 612 (FIG. 7b) and determining therefrom or "resolving" available references (typically in accordance with application support engine 614 of FIG. 7b or one or more of engines 621a-627~~525~~ of FIGS. 8a-8g~~8f~~ ~~available references~~. As shown, various "designation modes" can be provided as pertaining to a particular user, application, machine, etc. or be provided on a more global basis, as with the OEVI. Designation control 731 provides an overall designation control mechanism, as already ~~more generally~~ discussed. Explicit designation resolver 732, ~~Implied~~ implied designation resolver~~633~~733, operation resolver 734, data resolver 735 and application-machine resolver

~~726~~736more specifically provide for resolution of explicit, implied, tool, data and application/machine specific designations respectively. Command input resolver 736 provides for resolving available or applicable input commands, dictation or other data types (e.g. according to root, base, designation, extension, enhancement, user, group, user/group ID/status, history, context, and so on, for example, as was already discussed.

Resolution of explicit designations, such as a name or alias, can include determining where the designation might be located (e.g. in a local or remote address book, mail merge, remote control, etc.) and resolving no corresponding references or selecting from more than one reference (e.g. by user selection, preference, priority or selection rule). An implied designation mode provides implied designations including inclusive relative position (e.g. relative to the start or end of a grouping or a current item, including or excluding a current item), location (e.g. a next or input-corresponding data field or other presentation element), an inputting user, or an anti-alias (i.e. a pseudonym that specifically relates a generic identifier to one or more particular persons/items, typically in relation to an inputting or other user, such as "my mom", "X's <item>", etc. - see below).

The remaining resolvers typically provide, in addition to the resolution provided above, resolution of particularities with respect to operations, data or a given application, machine etc. For example, operation resolution can include identifying one or more particular machines, tools or modes of operation, particularly where more than 1 machine is used or specificities are input (thus, often utilizing app/machine resolver ~~735~~736 as well). Data resolution can include identifying remote data (e.g. the above-noted operation outside a pop-up window, etc.), as well as specific rules as to data position (e.g. whether a following non-abutting word is a "next" or "current" word.) Application or machine resolution can include applying specific parameters with respect to programs within a particular application, a particular program or other machines, processes, etc., among other examples (which should become apparent to those skilled in the art).

Cued input resolver ~~636~~738 provides for associating ~~an~~ later input as "cued" to an earlier one. For example, mixing music (or other applications) can require an extended setup, particularly where specified using voice commands; recognition and implementation of each one can further require an undesirably extended time period. Therefore, "cued" input resolver enables one or more prior "inputs" to be executed upon receipt of a "cue input" or more than

one such cues to be given. A given later-input cue can include, for example, an interpretable gesture (e.g. voice command) that can be implemented upon interpreting, timed interval, implicit/ explicit time/event, etc. or -typically more suitably- upon a "cueing" event that need only be recognized as occurring (e.g. moving a slider, clicking a button, etc.), where faster responsiveness is more desirable, or some combination. (Examples of suitable control mechanisms are also provided with reference to security/control in FIG. 8a and correspondence determination examples are provided in FIG. 8b.)

FIG. 7e illustrates how an exemplary enhancement engine 616 includes, for responding to an enhancement indicator, ~~(in addition to enhancement control 741 - see above),~~ repositioner 742, interface modifier 743, machine switcher~~644~~744, query engine 745, data carrier-filler~~646~~746, chooser 747, ~~and~~ machine controller 748 and documenter 749. Repositioner 742 provides for determining a typically pre or post command positioning of a presented or not presented designation indicator (e.g. curser/mouse pointer, present-next indicator, window/segment indicator, message response indicator, etc.) or a machine element (e.g. window/screen, window/screen segment, question-answer frame, conversation reference, etc.). Interface modifier 743 provides for adding/removing interface elements or causing the machine to otherwise operate in a non-standard manner (e.g. highlighting presentation elements, adding tools, altering window or other presentation sequencing, etc.); such modifications are providable, for example, via add-ins, applets, servlets, memory space modification, hooks, alternative hardware/object signaling -even macros, such as with the OEVI, among other more conventional or emerging mechanisms.

Of the remaining enhancement engine 616 components, machine switcher 744 provides for conducting coupling, information directing, machine linking, cueing, muting, and so on, as is discussed elsewhere herein. Query engine 745 provides for conducting querying of one or more users/devices, for example, as determined, coupled or directed by other execution engine 600 components. Data carrier/filler 746 provides for conducting data carrying/filling, as is discussed elsewhere herein. Choice/error feedback engine 747 provides for determining, generating or directing user/machine feedback. Machine controls 748 provides for determining or conducting machine control. Documenter ~~748~~749 provides for adding documentation or other data supplementing (e.g. in conjunction with command or data input, such as in identifying a current user, user edit, a confidentiality notice, implicit subject/object, etc.); data

elements other than text can also be similarly provided (e.g. graphics, such as a photograph or chart, video, audio, animation, etc.)."

The paragraph beginning **at page 67, line 27** has been amended as follows:

"Turning to FIG. 8a, an exemplary user engine 621(a) of user/use engine 621 (FIG. 6a) comprises user identifier 801, which typically operates in conjunction with security engine 622 and one or more of elements 802-808 to provide for identifying one or more users. Controller identifier ~~702~~802 provides for identifying a user in accordance with a particular controller (e.g. a microphone or localizing combination of microphones). Currently, such identification is more likely applicable once a user-controller correspondence has been established, such as after identification by another user or the system, assuming such association can be established. (New devices might more suitably provide for initial and ongoing user/device identification, such as when activated -e.g. speaking into a microphone or moving a pointer; current devices can, however, also be utilized via polling of the device/device-connection, receiving an interrupt/connection indicator, localizing or other mechanisms.)"

The **3 consecutive** paragraph beginning at **page 68, line 25** have been amended as follows:

"Once associated, each user can, for example, be identified for multi-user data entry (e.g. question-answer, workgroups, etc.) that can further provide automatic annotation ~~or~~ in a static or modifiable manner (e.g. corresponding to application, local/remote machine, one or more user/system parameters, ~~formatted~~ formatting, etc.). Control or flexible secure control, data entry or "reporting" or still further capabilities can also be provided to one or more local or remote users.

Of the remaining user engine 621(a) elements, status identifier ~~805~~806 provides for associating an identified user, such as given above, with various status-determinable criteria. Upon receipt of a user indicator and attempted input, status identifier determines whether such input is allowable (e.g., in conjunction with security engine 622 of FIG. 6a) and returns a corresponding status indicator. Such "status" can include a single user status for accessing various system controls or information (e.g. security below). Status identifier ~~805~~806 also provides for group-related status indicators, such as where one user is an officiator of a conference, an uninterruptible or conditionally interruptible participant, etc.). Machine-location

identifier 807 provides for associating a user and a user-information storing machine (e.g. user preferences, speech files or other control parameters or other of user's remotely stored information. Finally, user information engine 808 provides for ~~retrieve~~retrieving a default, application/machine or other set of user-information from storage or a "located" remote machine.

FIG. 8b illustrates how an exemplary use engine ~~616~~621(b) of user/use engine 621 (FIG. 6a) includes use analyzer 811 (see controllers/analyzers above), application engine 812, machine engine 813, capabilities engine 814, extended capabilities engine 815 and location engine 816, each of which responds to a requesting element by providing use-related information or conducting one or more use related operations. Application engine 712 provides for determining application-related parameters in accordance with a current or next application (e.g. in an application-transitional context), such as defining user/application parameters to retrieve information from a local or remote source or in executing commands relating to a particular application, and machine engine 813 similarly provides for determining machine-related parameters. Capabilities engine 814 provides for further determining enabled/disabled non-current operabilities or enabling or disabling them. Extended capabilities engine 815 further provides for determining/facilitating enhanced or combined-machine operabilities, as with application or machine engines 812, 813. Finally, location engine 816 provides for determining (e.g. via history, machine polling, etc.) a current or persistent pointer location (e.g. mousepointer, curser, etc.) in accordance with which commands can then be executed (see location generally, cued operation and other examples above)."

The **2 consecutive** paragraphs beginning at **page 70, line 4** have been amended as follows:

"Analyzer-filters ~~823-825~~822-826 facilitate security more particularly related to the freedom provided by conversance or otherwise more "free-form" I/O. That is, rather than treating user/ machine input treated as generic, analyzer-filters ~~823-825~~822-826 provide for determining received information content or content types and, upon determining such information (e.g. a particular designation, specificity, confidential data element, etc.), provide for correspondingly removing, modifying or replacing the input element(s). Such analysis/determining can, for example, be conducted via parsing input, stored or for-output information respectively, and comparing the information to a stored or otherwise ascertainable

element set; such element or elements can be replaced, deleted or modified in accordance with corresponding elements or other processing (e.g. users, purposes, contexts, tasks, applications, machines, providing feedback, etc.) (Other suitable mechanisms can also be used.).

Analyzer-filters ~~823-825~~822-826 can, for example, be used to remove a company, individual or product indicator, locator or parameter; confidential element; tool use; contextually significant information; etc. as spoken into a cell phone for storage or re-transmission (e.g. of biometric data, from remote conference members, to a generally accessible database or from a private one, to a remotely initiated correspondence addressed to a particular user, group or machine, etc.). A suitable feedback can also be provided to indicate such processing or provide a warning as to the receipt of such data."

The paragraph beginning **at page 70, line 31** has been amended as follows:

"FIG. 8d illustrates how an exemplary reliability engine 623 comprises reliability analyzer 831, purpose/use analyzer 832, colloquialism engine 833, ~~inflexion~~inflection engine 834, conversance reliability analyzer 835, application-location progress analyzer 836 and user(s) preference analyzer 837. Reliability analyzer 831 provides for determining from received input, interpretation/ utilization possibilities corresponding thereto in conjunction with the remaining elements. Purpose/use analyzer ~~835~~832, for example, provides for including commands or data consistent with (or excluding commands/data inconsistent with) a determined purpose or use (e.g. application, machine, etc.); note that such purpose or use might further determine a particular manner of operation (e.g. applying user parameters, vocabularies, etc.), or might merely narrow command recognition possibilities (as a command/data, particular command/data, etc.). FIG. 12b, for example, shows how dictation or other data or types or other particular instances thereof may, in a given case, be excluded from entry into window 1202a but included for entry into coupled or other windows 1202b-d. Colloquialism engine 833 provides for determining a colloquialism or other typically vocabulary selection alternative consistent with other reliability metrics (e.g. charting, use of abbreviations, formal letters versus familiar, etc.). ~~Inflexion~~Inflection engine 834 provides for determining, from an input inflexion (e.g. pitch, speed, direction, pressure, etc.), a corresponding more or less likely command/data; a raised voice might, for example, indicate a mistake by the system such that a correction machine might be invoked, or other correspondences might also be utilized (see above)."

The paragraph beginning **at page 71, line 31** has been amended as follows:

"FIG. 8e illustrates how an exemplary conversance engine 624 includes conversance analyzer 841, context engine 842, purpose-task analyzer 843, rhythm engine 844, continuity engine 845 and use interaction analyzer 846. Conversance analyzer ~~741~~841 provides for conversant utilization (e.g. for greater reliability with lesser or non-conversantly structured commands) or further reliability in accordance as is achievable in accordance with conversant aspects. It will be appreciated, for example, that context, tasks, etc., can be identified and utilized as traditional or artificial intelligence metrics during execution even where a word or structure is otherwise unsupported. Accordingly, context engine 842 provides for determining a current or successive context, and purpose-task engine 843 provides for determining a current or successive purpose/task (or trend thereof). Continuity engine 845 determines a likelihood of a continuing conversance trend or likely consistent "next" command/data element, for example, corresponding to recent history. Use-interaction analyzer 846 provides for tailoring the above element results in accordance with a particular approach, environment, machine, etc. (e.g. providing for a response consistent with a telephone, OS, question-answer interface, etc.)"

The paragraph beginning **at page 72, line 20** has been amended as follows:

"FIG. 8f illustrates how an exemplary history engine 625 comprises user preference settings engine 851, conversant preferences engine 852, tendencies engine 853, progress engine ~~754~~854, concatinator-splitter 855 and predictor reliability engine 856. Each such component provides for facilitating use of recent/trend or further-extending history information. User preference settings engine 851 provides for user-specific, or further, user determinable preferences for evaluating history information. Conversant preferences engine 852 provides for evaluating or modifying conversant aspects corresponding to one environment, application, machine, etc. differently ~~than~~from another or further with respect to one or more users (e.g. by individual or group personage, status, etc.). Tendencies engine 853 and progress engine 854 similarly provide for evaluating or modifying tendency or progress parameters respectively and thereby affecting response by the system to such factors (see above)."

The paragraph beginning **at page 73, line 15** has been amended as follows:

"~~Finally,~~ FIG. 8g illustrates how an exemplary information communicator 627 -the remaining support engine element of the present embodiment- includes communication engine 861, locator 862, intra-machine negotiator 863, inter-machine negotiator 864 and user negotiator 865, each of which facilitates local or remote communication of information. Communication engine 861 provides for ~~communication~~communicating information within a host machine or included machines, or with external machines. Locator 862 provides for identifying such machines (e.g. an external or "remote" user machine containing retrievable user speech or other information automatically upon user identification or otherwise). Intra-machine negotiator 863 provides for interfacing with a host machine or included machines, while extra-machine negotiator provides for interfacing with remote machines and user negotiator 864 provides for interfacing with a user (e.g. supplying such information to communication engine 865 in accordance with a location or other machine/element identifier returned by locator 862 to communication engine 861)."

The paragraph beginning **at page 76, line 21** has been amended as follows:

"Note that the methods disclosed herein are applicable not only to voice-enhancement of existing applications and voice-enablement of new applications, but also where underlying or speech-based features are added or modified. It is possible in each case that underlying application/interface elements might be incapable of operations that are sufficiently consistent with user tasks or other voice-interfacing aspects; if so and capabilities cannot be added using available tools, then a gap will exist with respect to affected tasks. This is unfortunate but not necessarily critical; such a gap might, for example, have lesser overall importance or be rendered less noticeable to a user. In this sense, initial distillation facilitates identification of tasks, which tends to facilitate an interface that is less affected by reflexively applied and often contrary to conventional approaches or solutions."

The paragraph beginning **at page 78, line 28** has been amended as follows:

"A particularly surprising discovery is that perhaps even more important than the specific verbiage used in communicative voice-commands are balance and consistency. While an ideal interface might provide an integration of voice-commands with other interface elements, similar benefits are not necessarily achieved by accommodating existing application/interface elements

with targeted voice-commands. Rather, interface efficiency is found to be enhanced by causing the underlying application/interface to respond in accordance with voice-interface optimization, such as that disclosed herein; to the extent that this is not possible, a localized conflict is often preferable to a forced existing application/interface accommodation. In some cases, this might mean adding additional capability. For example, greater efficiency has been achieved in certain cases by assuring that capabilities relating to tasks supportive and complimentary to an included task are also included (i.e. balance); greater efficiency has also been achieved by assuring that a task supported in one instance is similarly supported in other applicable instances (i.e. consistency). The conversant interface of FIG. 13a, for example, enables addressing to be conducted in a similar manner –here using nth terminology- in conjunction with otherwise disparate underlying interface portions 1301a-c. The conversant interface of FIG. 12g, similarly enables a user to conduct – in a consistent manner - additive item selection in conjunction with underlying interface portion 1270 and additive lookup criteria entry in conjunction with underlying interface portion 1270a."

The paragraph beginning **at page 80, line 4** has been amended as follows:

"An overlayed window-switching method is employed using NS tool activation of underlying GUI elements to enable ~~relatively~~a relatively highly balanced and consistent user experience among all three of the above windows. That is, recitation of appropriate "addressing" voice-commands cause the display of New Message window, Select Recipients window or a combination of the Select Recipients and Find People windows. (See, for example, FIG. 13a.) Thereafter, common voice commands are provided for all three windows."

The paragraph beginning **at page 80, line 29** has been amended as follows:

"~~A~~ FIG. 13c further illustrates an example of how a user can also invoke various voice-commands to exit the Find People window. A user can "close" the window 1332a (e.g. as with conventional OE and NS controls) and return to the Select Recipients window. The user can also accept or reject addressing by apparently directly using the revealed Select Recipients "OK" and "Cancel" buttons 1333b. Implementationally, the Find People window is closed (e.g. using an escape key equivalent or "close"), and the respective Select Recipients OK or Cancel button is actuated; the user is then preferably returned to the message portion of the New

Message window, and optionally, to either the last cursor position or a "marked" position 1333d. Alternatively, commands are also provided (here and in the Select Recipients window) for sending the current email message, which returns to the New Message window and selects "Send" 1333e-g. (Note that other options can also be provided in a similar manner)."

The paragraph beginning **at page 81, line 16** has been amended as follows:

"Results 1 and 2 suggest that the effect of balancing ~~superceded~~superseded its apparent conflict with prior experience using GUI-based controls, while greater consistency heightened the effect (as given by results 3 through 5). Additionally, each of the task-based voice commands, existing-interface modifications, balancing and consistency appeared to actually guide voice-command expectations, making the commands more intuitive and thus more efficiently utilized. (Even the not-represented send command 1333e of the Find People and select recipients windows quickly became almost automatic.) In retrospect, an overall user experience was also apparently achieved of telling an assistant to address an email message, to add recipients and/or delete them (e.g. given a change of mind), and to accept or ignore the changes."

The paragraph beginning **at page 84, line 28** has been amended as follows:

"As a further example, a conventional view would suggest that the following all provide unique emailing capabilities and each is therefore conventionally treated as such: highlighting new email messages, managing contacts, finding related existing messages, addressing, message inserts (e.g. attachments) and the like. It is found, however, that they can also be more generically viewed together as relating to handling groupings of things (which was eventually ~~finally~~ characterized as a conversational context)."

The **2 consecutive** paragraphs beginning at **page 87, line 15** have been amended as follows:

"Data carrying broadly refers to locating, storing and then reusing data from one location as if entered in another location. ~~For example,~~FIG. 12d, for example, shows how reciting a "Find more messages" voice-command from within the OE Outlook window causes messages

from the sender of the current message to be listed in the Find messages window. Ideally, all aspects of messages would be available from within the Outlook window, perhaps via polling (as would similarly be provided for relevant aspects of other item groups in other circumstances). However, because email addresses are unavailable in the Outlook window, control is shifted to the Preview Messages window 1241 and the sender's email address is copied (using NS mouse commands for selection and a copy keyboard equivalent). Control is then shifted to the Find Messages window 1243 (by selecting a menu item), where the copied data is pasted into the "To" field 1244, extraneous data not included within the raw email address is removed and the "Find" button is actuated 1247. (The Find more messages command is also implemented in a similar manner from within Preview window, but without data carrying).

Permutations of the Find more messages commands are similarly implemented. In this case, speaking a command that also includes criteria and/or message aspects results in the same being either deposited or used in selecting available options within the Find Messages window. For example, flagged messages and/or messages with attachments permutations can be reflected by selecting these attribute options 1246; messages to or from a recipient can further be reflected by copying and pasting of appropriate criteria into respective Find Messages fields 1244. Assuming that modifications such as those given above are implemented, exchanges can also be reflected by copying sender *and* recipient data, and then pasting the data to the appropriate to and from (or modified) fields, and so on. (Similar carrying of data from the OE Preview window to the Find Messages window are also used from within the Find Messages window and the same and similar voice commands are provided.)"

The paragraph beginning **at page 90, line 30** has been amended as follows:

"Communicative factors also appear to be heightened and greater efficiency enabled by a less interactively ~~conversational~~conversant interface based on "telling an assistant what to do" (hereinafter "instructional conversation"). The OE voice-interface, for example, illustrates how such a system of commands can be used to create an unobtrusive yet more effective awareness that commands are being used to effectuate control, and in a particular way. It also demonstrates how such effectiveness can more efficiently achieved using (and despite) such factors as a PC, aging operating system, directed application, OE and NS. Among other

advantages, a user will further tend to think and speak in a more predictably and guideably structured and verbalized manner as compared with other potential systems. Further capabilities provided by more capable machines, tools and other factors will likely enhance a user experience to an even greater extent for other forms of conversational interfacing, and even more so for instructional conversation."

The paragraph beginning **at page 91, line 25** has been amended as follows:

"Voice-commands are preferably formed in accordance with the above analyses as comprising a root-command and an optional enhancement-portion. The *root-command* most often corresponds with a basic task, and most often discretely comprises at least one process and one or more designated objects of that process. A simple root-command might, for example, roughly correspond with two or more conventional commands, but with the process and ~~task~~object order reversed for facilitating better conversational flow (e.g. "Send email" versus "Select email" followed by "Click send"). Further voice-commands might correspond with conventional commands only with respect to speech-tool invocation of conventional controls (as with NS), corresponding results, or not at all (e.g. using other than conventional control invocation). Command *enhancements* most often correspond with permutations of the basic task; however, they can also include elements normally included within the basic command or a more definitive statement of the basic task, among other possibilities."

The paragraph beginning **at page 94, line 17** has been amended as follows:

"Inter-item designation, for example, can be used where a reference item can be distinguished as a user progresses through an identifiable group of items. In this case, items are preferably designated according to their position relative to a movable or transferable current item reference and voice-command verbiage is selected accordingly. That is, verbiage is included for designating the current item as well as one or more other relatively positioned items both alone and in conjunction with the current item. More preferably, verbiage is selected for designating at least adjacently positioned items including: (1) each of the preceding, current and following items, (2) each of more than one preceding and more than one following items, and (3) each of the current item plus one or more following items and (to a lesser extent) the current item plus one or more preceding items. Such verbiage more preferably references

the current plus preceding and current plus following items of case (3) in a distinctively different way from cases (1) and (2). See, for example, FIG. 12f-h."

The paragraph beginning **at page 96, line 6** has been amended as follows:

"The OE voice-interface employs feature-based designation in instances such as with the Find People window ("FPw"), where a (highlighted) current item appears to exist but is actually reset following entry of an addressee. Two structures and new verbiage are utilized (in accordance with testing) to produce the above-discussed addition and deletion of addressees. See, for example, FIG. 13a. More specifically, the structure

<process> <nth> <optional connector> <optional nth>,

is used for adding one or more addressees, wherein "nth" indicates a position relative to the beginning of a list, all items or the last item in a list, producing commands such as "To 1st "; "Cc 2nd and 3rd "; and "Bcc 4th through last." The structure

<process> <discrete or implied nth> <group designation>

is further used for deletion (or undoing) one or more addressees in such commands as "Delete To" (which implies the last addressee added to the To field of Find People). It is also used with permutations such as "Delete last To", "Undo last 3 Cc" for greater consistency with other compatibly metered and/or worded voice-commands. (The second structure and verbiage also represent somewhat unique instances of intermittent-designation as discussed below.)"

The paragraph beginning **at page 97, line 24** has been amended as follows:

"The OE voice-interface, for example, conducts predictive repositioning and designates new messages within the OE Outlook window Message list as follows. It is presumed that a user will handle successive grouped items, and particularly new messages, from an apparent list-beginning toward a list-end (e.g. from top to bottom); however, a user might also return to a prior message, handle the same message(s) in more than one way or pause/disrupt message handling and perhaps return to it again later. Accordingly, with few exceptions, following

handling of prior messages, the previous current message is again (automatically) designated as the new current message, and otherwise the message immediately following a last-handled message (i.e. furthest down a list) is designated as the new current message. See, for example, FIG. 12f."

The **2 consecutive** paragraphs beginning at **page 100, line 2** have been amended as follows:

" "External control," as used herein, provides for manipulating displayed or not-displayed controls associated with a foreign item intermittently while handling a current item or items. For example, while handling messages, a user might want to scroll up or down the folders list to reveal potentially applicable folders (e.g. in which to move a message). See, for example, FIG. 12c.

"Simple paging," as used herein, further provides for affecting one or more foreign items intermittently while handling a current item or items. For example, a user might want to further open a folder that is displayed as a result of the above external control. "Complex paging," as used herein, still further provides for designating and/or affecting one new item with reference to another new item while handling a current item or items. For example, a user affecting a message within one folder might want to similarly affect a further message in another folder. In other cases, complex paging might also be used as a type of conversational relative addressing, communicative linking, etc. among a variety of similar or dissimilar items, types, applications, etc. (See, for example, FIGS. 12e-1-12e-2.)"

The paragraph beginning **at page 101, line 1** has been amended as follows:

"Among other methods, intermittent-context can be effectuated by adding an implied or ~~discrete~~explicit new item reference to existing voice-command structures for designating new items and/or item groups. For example, intermittent-context methods are typically used in the OE voice-interface in conjunction with the above relative-positioning according to the structure

<process> < (local) designation> <optional number of items> <(extended) designation>,

wherein the local designation is preferably the same as the above non-intermittent designation; and the extended designation is an item/group reference. An extended designation can, however, also be positioned elsewhere in accordance with communicative factors (as with permutations)."

The paragraph beginning **at page 105, line 3** has been amended as follows:

"The above window referencing methods can also be used to extend the capability of an underlying application even where conventional interfacing and speech tools are utilized. ~~For example~~FIG. 14, for example, shows how issuing a "Merge and reply to these three" command from the OE Outlook window messages list or a "Merge and reply to these three messages" from the folders list first creates a new reply message including the received current message. Thereafter, control is switched to the Outlook window and each of the two following adjacent messages are merged using window cycling. More specifically, a first merged message is opened in the Preview window, its contents are copied and the Preview window is closed, thereby switching back to the Outlook window. Control then switches to the reply message and the copied contents are pasted. The copy and paste steps are then repeated for the remaining message (also inserting a graphical separator between messages, such as a line). A similarly implemented "Merge and forward" feature is also provided."

The **2 consecutive** paragraphs beginning **at page 107, line 9** have been amended as follows:

"The OE voice-interface implementation of random designation, for example, preferably incorporates the above relative-item designation command structure and verbiage with a new continuation command type having the structure

<connecting term> <non-continuation intermediate or non-intermediate command>.

In this instance, the connecting term is "and", and the resultant commands are referred to hereinafter as "And commands." The non-continuation command portion is further typically a "select" or "move" command. See, for example, FIG. 12g. Processing-based And commands (e.g. "And flag that") might also be provided; however, a user most often tends to expect

separate processing capability and a strong need for including such commands has not yet been demonstrated. (Also interesting is that while a random designation might also begin with an And select command performing in its usual manner, this possibility has not occurred during testing.)"

The paragraph beginning at **page 109, line 6** has been amended as follows:

"Thus, several improvements would be desirable. For example, at least a displayed item numbering should be added in a simple, intermediate or complex form should be provided (e.g. respectively by item, rows and columns as with spreadsheets, or a separated folder/file hierarchy both visually and alphanumerically). See, for example, the enabled use of nth numbering 1314a-c in conjunction with windows 1301a-c of FIG. 13a. Another possibility is to simply abandon aspects of the traditional approach entirely and provide a graphical file drawer, folder, sub-folder and file paradigm supported by graphical and other non-speech multimedia elements (e.g. 2-dimensionally separated indicators, e.g., 1315 of FIG. 13a, a 3-dimensional environment, etc.). Using such more robust capabilities, efficiency could be greatly improved in accordance with intermittent-designation, integration of variable names and/or other methods."

The paragraph beginning **at page 112, line 17** has been amended as follows:

"In the OE voice-interface, special-contact designations are predetermined as command designations (as necessitated by NS-tools) and matching entries within the "display" fields of various OE contacts created for this purpose. Designations are primarily of the structure <designation indicator> <reference> <optional data>, and include such designations as "my secretary", "my boss" and "my cousin" ~~(See Parital Command list)~~. The term "my" primarily serves 2 purposes: to distinguish the reference as relating to a special contact (e.g. versus a name or alias); and to provide grammatical flow and rhythmic consistency. The term "secretary" or "boss" distinguishes the particular designation from other types. That is, presuming the user has but one secretary or boss at any given time, whomever constitutes the current secretary or boss can be specifically designated by the reference; that person's contact and miscellaneous information can also be entered as contact information for that reference."

The **5 consecutive** paragraphs beginning at **page 113, line 17** have been amended as follows:

"Special-identifiers can also be used in unique ways in combination with other aspects of the invention. ~~For example, the accompanying figures illustrate~~ FIG. 12a, for example, illustrates how permutations, connecting commands and partial/joined commands can be used to greatly increase user efficiency in creating and addressing messages. As shown, a user within the OE Outlook window (or other windows) can initiate a new email using the voice commands "Send an email" or "Send an email to" 1201a. The "Send an email" (Root) command swtiches from a current window, initiates a New Message window, enters the "Re" designation (see above) and switches to the "To" field. A user may then recite additional voice commands for addressing the email and/or performing other tasks. Alternatively, a user can recite a "Send an email to" command (which is also a Root command) to produce other results. The word "to" is used as a further mental, verbal and programmatic connector included in all natural cases (e.g. it is not "forced" on a user, but is included singly or as an alternative in all applicable cases). Surprisingly, the connector "to" maintains an experiential continuity both directly and in an anticipatory manner, as illustrated by the remaining alternatives. (It also turns out to be used in an extremely intuitive manner for selecting each of the alternatives via inclusion or exclusion of specific verbiage types.)

In a second alternative, a user can recite a "Send an email to <special-designation>" permutation 1201b, such as in sending an email to a special-contact. Here, stating the special contact performs the "Send an email" command and enters the associated addressee; upon checking the addressee entries (which would otherwise be performable automatically, but here requires a further voice-command), OE will ~~confrim~~confirm the entry as listed in the address book, provide a list of matching potential addressees from the address book or suggest alternatives and enable entry of a new contact. (Adding a particular reference to special-contacts would also increase efficiency should such features be adopted.) As was already noted, ~~the word~~ "my" or "X's" (e.g., John's or some other relationship) may be used to indicate a special contact.

In a further alternative, a user can recite a partial command of "Send an email to the" or a joined command of "Send an email to the <folder name>" 1201c. Such commands invoke a "Send an email to" command and further switch control to the OE Select Recipients window

(by virtue of the word "the", "a" or "an"). The partial command further opens the contact folder list, thereby enabling a user to "complete the command" by reciting a folder name of which he need not have previously be aware. Surprisingly, the user anticipation created by the partial command creates an experience in which rhythmic flow (or at least flow continuity) is largely preserved. The joined command invokes a "Send an email to the" command and further selects a folder, closes the folder list and (at present) places the user at a data entry field. A user than then make a more specific selection (if needed) for listing appropriate contacts.

In a still further alternative, a user can recite the joined command "Send an email to <field or window name> 1201d, which invokes a "Send an email" command and further transfers control to an apporiate window and (optionally) an appropriate field within that window. More specifically, this set of commands transfers the user to the Select Recipients or Find People windows. (Other commands invoke address book windows, pages and fields in a similar manner). The term joined (or "transitional") is used to express the rhythmic/context modification enabled by such a command. For example, a user might recite "Send an email to phone number", which not only selects the phone field for subsequent user entry of a telephone number, but also effectively changes the context and ~~rhymic~~rhythmic meter. That is, the new context of criteria based extraction also enables an intuitive shift in rhythmic flow appropriate to the context (ignoring enabled alternatives). Here (as with similar address book instances) the rhythmic flow is much simpler. The user can simply state a field and then separately state the data to be entered in that field (e.g. "555-5555"... "Name"... "John", etc.).

While necessitated due to NS inability to use variable new data, clearly the abilities to change context and rhythm, and further to use partial commands would remain useful alternatives even when the NS deficiency is corrected in providing a complete and thus intuitive and efficient conversational experience. Additionally, other interface aspects are also used in both the address book and Find People cases whereby an arrow (here, a mousepointer) is used to indicate an entry field. However, continued use of the pointer is maintained only within the address book. Preferably, such use is determined based on the complexity of the display as well as the tendency of the feedback utilized to either clarify or distract a user, among other factors."

IN THE CLAIMS:

Please amend *EXISTING* claims 1-4 and add new claims 5 through 44 as follows.

1. (Currently Amended) An interfacing method conducted by one or more machines, the machines including a host computing device and one or more hosted machines, comprising:

receiving voice information corresponding to at least one machine user; and

processing the voice information, the processing including:

determining whether the voice information includes command information; and if so, then:

determining one or more use-based objectives corresponding to the ~~machine~~ voice information;

determining one or more specificities corresponding to the command information; and

determining a conversant ~~commands~~ command execution corresponding to the use-based objectives and the specificities.

2. (Original) An interfacing system formed according to the method of claim 1.

3. (Currently Amended) An interfacing system comprising one or more machines, the machines including a host computing device and one or more hosted machines, the system comprising:

means for receiving voice information corresponding to at least one machine user; and

means for processing the voice information, the processing including:

determining whether the voice information includes command information; and if so, then:

determining one or more use-based objectives corresponding to the ~~machine~~ voice information;

determining one or more specificities corresponding to the command information; and

determining a conversant ~~commands~~ command execution corresponding to the use-based objectives and the specificities.

4. (Currently Amended) A computer readable medium having stored thereon computer code for causing a computer to perform the steps of:

receiving voice information corresponding to at least one machine user; and

processing the voice information, the processing including:

determining whether the voice information includes command information; and if so, then:

determining one or more use-based objectives corresponding to the ~~machine~~ voice information;

determining one or more specificities corresponding to the command information; and

determining a conversant ~~commands~~ command execution corresponding to the use-based objectives and the specificities.

5. (New) The method of claim 1, wherein the voice information comprises at least one of: a conversant voice command recitation of one or more users, a non-conversant voice command recitation of one or more users and monitored speech of one or more users.

6. (New) The method of claim 1, wherein the determining one or more use-based objectives comprises determining that at least one voice information portion of the voice information corresponds to at least one of a user task and a user goal.

7. (New) The method of claim 1, wherein the determining one or more use-based objectives comprises determining that at least a voice information portion of the voice information corresponds to at least one of a group task and a group goal.

8. (New) The method of claim 1, wherein the determining one or more specificities includes determining one or more explicit specificities as corresponding to one or more explicitly recited voice information portions of the voice information.

9. (New) The method of claim 1, wherein the determining one or more specificities includes determining one or more implied specificities that the voice information does not explicitly include.

10. (New) The method of claim 1, wherein at least a portion of the processing is conducted in accordance with one or more of a user identification, a user security, a user status and a user group status, the group status corresponding to at least one of a workgroup, a moderated workgroup, membership, officiator, participant, interruptability and member-grouping.

11. (New) The method of claim 1, wherein:

the processing further comprises associating one or more of a user location, a user positioning, user movement, user gaze, user non-verbal gesture, and user non-verbal gesture inflexion with the voice information; and

at least a portion of the processing is conducted in accordance with the one or more of a user location, movement, non-verbal gestures and non-verbal gesture inflexions.

12. (New) The method of claim 1, wherein at least a portion of the processing comprises determining one or more machine portions of the one or more hosted machines for effecting the one or more use-based objectives.

13. (New) The method of claim 1, wherein at least a portion of the processing is conducted in accordance with one or more of a machine group, security of a machine use, a machine location and a correspondence of a machine with at least one of a user, user group, security and status.

14. (New) The method of claim 1, wherein at least a portion of the processing is conducted in accordance with one or more of a machine portion use, a machine portion purpose, an availability of one or more machines and an availability of one or more machine portions of one or more machines.

15. (New) The method of claim 1, wherein at least a portion of the processing is conducted in accordance with one or more of a context, a conversant context, an interaction, an approach and a scenario.

16. (New) The method of claim 1, wherein the processing further comprises:

determining at least one likely further use-based objective that may be determined in accordance with further received voice information; and

conducting at least a portion of the processing in accordance with the at least one likely further use-based objective.

17. (New) The method of claim 1, wherein the processing is conducted in accordance with at least one of a processing history, a user habit and a user tendency of at least one user.

18. (New) The method of claim 1, wherein at least a portion of the processing is conducted in accordance with a content characterization, the content characterization including at least one of an information type, an information use, an information application and an information purpose.

19. (New) The method of claim 18, wherein the information type is selected from a group including commands, data, biometric data, dictation, and specific data type.

20. (New) The method of claim 18, wherein the information type is selected from a group including silence, misstatement, mis-recitation, private information and confidential information.

21. (New) The method of claim 18, wherein the information use is selected from a group including personal information and business information.

22. (New) The method of claim 18, wherein the information application is selected from a group including charting, home control, calendaring, vehicle operation, communication, multimedia production, media presentation and document production.

23. (New) The method of claim 18, wherein the information purpose is selected from a group including a particularized objective and a subject matter of a user to which at least one of a command portion, a data portion, a dictation portion is determined to be directed.

24. (New) The method of claim 18, wherein the information purpose is selected from a group including entering or reviewing form data, addressing a subject, orchestrating, conducting production console operation, controlling effects, indicating media presentation elements, operating an entertainment system portion, operating entertainment receiver mode, faxing, emailing, calling, conferencing, preparing or reviewing a formal or familiar document, preparing or reviewing annotation, preparing or reviewing a multimedia portion, preparing or reviewing a clause and preparing or reviewing a document type, section or section type.

25. (New) The method of claim 1, wherein the processing is conducted in accordance with a usage ("expression characterization") corresponding to at least a portion of the voice information.

26. (New) The method of claim 25, wherein the manner of expression is selected from a group including language, dialect and colloquialism, inflexion, biometrics, physical gesture and non-speech expression.

27. (New) The method of claim 1, wherein the processing comprises at least one of enabling and disabling processing of at least a portion of the voice information in accordance with at least one of a content characterization and an expression characterization.

28. (New) The method of claim 1, wherein the processing comprises at least one of trapping, muting, modifying, substituting for and directing transmission, including non-transmission, of a voice information portion in accordance with at least one of a content characterization, an expression characterization, a reciting user, a target user, a machine portion and one or more of the specificities.

29. (New) The method of claim 1, wherein the processing comprises determining at least one of a local vocabulary portion, a remote vocabulary portion, a non-vocabulary recognition component, an interpretation and an output format alternative in accordance with at least one of a content characterization and an expression characterization.

30. (New) The method of claim 1, wherein the processing comprises determining at least one of an operational mistake, corrective action and implicit user assisting in accordance with at least one of a content characterization and an expression characterization.

31. (New) The method of claim 1, wherein the processing comprises:

determining that a portion of the voice information corresponds with an anti-alias, the anti-alias comprising an anti-alias designation indicating at least one specific target of the anti-alias; and

resolving the anti-alias.

32. (New) The method of claim 31, wherein:

the resolving the anti-alias comprises determining at least one specific target in accordance with at least one of a current class membership, a current title and a currently performed function;

the at least one specific target and the at least one of a class, title and function correspond with one or more of explicit and implicit specificities, the one or more specificities further corresponding to at least a portion of the voice information; and

the voice information corresponds with one or more recitations.

33. (New) The method of claim 31,.wherein:

the anti-alias designation indicates at least one of a target classification designation and a source of target resolution information in accordance with which the anti-alias may be resolved; and

the anti-alias further comprises one or more of:

at least one anti-alias indicator indicating at least one of a presence of the target anti-alias designation, an association of the target anti-alias designation with at least one

specificity, target resolution information in accordance with which the anti-alias may be resolved and a target source of resolution information in accordance with which the anti-alias may be resolved; and

at least one specifying indicator indicating one or more specificities in accordance with which the anti-alias may be resolved.

34. (New) The method of claim 33, wherein at least one of:

the anti-alias designation indicates possession;

the at least one anti-alias indicator indicates possession;

the at least one specifying indicator indicates one or more name portions; and

the at least one specifying indicator indicates one or more title portions.

35. (New) The method of claim 1, wherein:

the determining a conversant command execution includes designating at least one machine portion for executing at least a portion of the voice information;

the designating is conducted in accordance with at least one of a not explicitly stated ("implied") specificity and the use-based objective, thereby enabling one or more of feedback corresponding to the transition or non-transition, completion of a designation objective and preparation corresponding to a likely successive user recitation; and

wherein at least one of:

the designating causes the executing to be conducted by invoking operabilities of a currently designated ("current") machine portion and the designation to then transition to a not currently designated ("non-current") machine portion;

the designating causes a designation of a current machine portion to transition to a non-current machine portion, causing the executing to be conducted by invoking operabilities of the non-current machine portion, and the designation to then remain as the non-current machine portion or further transition to a different non-current machine portion; and

the designating causes a designation of a current machine portion to intermittently transition to a non-current machine portion, the executing to be conducted by invoking operabilities of the non-current machine portion

and the designation of the non-current machine portion to then transition back to the current machine portion.

36. (New) The method of claim 35, wherein the designating is further conducted in accordance with at least one of an operational history, a user habit and a user tendency.

37. (New) The method of claim 35, wherein the determining a conversant command execution further comprises conducting determining a preparation for a successive recitation, the preparation including:

determining a not explicitly recited first designated machine portion for providing data and a not explicitly recited further designated machine for receiving the data;

causing the execution by the first designated machine to provide the data; and

causing data carrying comprising inputting the data into the further designated machine portion.

38. (New) The method of claim 1, wherein at least one of the machines provides for presenting a graphical user interface (GUI) portion and at least one of the hosted machines comprises at least one of an operating system portion, an application program portion, a window, a window pane, a data portion, a control interface portion and data entry field portion.

39. (New) The method of claim 1, wherein:

the voice information includes previously received voice information and currently received voice information;

at least one of the determining a use-based objective, the determining one or more specificities and the determining a conversant execution includes determining that at least a portion of the currently received voice information corresponds with at least a portion of the previously received voice information ('determining a correspondence");

at least one of the determining a use-based objective, the determining one or more specificities and the determining a conversant execution includes processing at least a

portion of the previously received voice information and the currently received voice information as a discontiguous recitation of a use-based objective.

40. (New) The method of claim 39, wherein at least one of:

the determining a correspondence determines that the previously received information and the currently received voice information correspond with an interrupted single recitation;

the determining a correspondence determines that the currently received information at least partially completes the previously received information by including one or more corresponding specificities having a same input type characterization;

the determining a correspondence determines that the currently received information at least partially completes the previously received information by providing one or more corresponding specificities having a different input type characterization;

the determining a correspondence determines that at least one of the previously recited information and the currently received information comprises a partial recitation; and

the determining a correspondence comprises determining that at least one of the previously received voice information and the currently received voice information includes a linking indicator.

41. (New) The method of claim 39 wherein the conversant execution includes:

conducting independent processing corresponding to at least a portion of the previously received information;

preserving the independent processing if the currently received voice information includes a linking indicator; and

modifying at least a portion of a result obtained in accordance with the independent processing if the currently received voice information does not include a linking indicator.

42. (New) The method of claim 41, wherein the use-based objective includes at least one of:

designating an item group of items that may include discontiguous items, the designating enabling the item group be similarly processed in accordance with subsequently received voice information;

designating a criteria group of criteria, the designating enabling processing to be conducted in accordance with the criteria group;

inputting a criteria group, the inputting enabling processing to be conducted in accordance with the criteria group; and

designating at least one criteria and inputting one or more criteria, the designating and inputting enabling processing to be conducted in accordance with the designated and input criteria.

43. (New) The method of claim 1, wherein at least one of the determining a use-based objective, the determining one or more specificities and the determining a conversant execution includes at least one of:

determining that the voice information includes a cueing indicator indicating that the voice information comprises a cued command and at least a portion of processing of the voice information should be forestalled unless a corresponding cue initiating trigger is subsequently received; and

determining, if a cue initiating trigger is received, that processing of a corresponding cued command should be initiated.

44. (New) The method of claim 43, wherein the cue initiating trigger includes at least one of:

voice information including a cue initiating indicator indicating that processing of at least a portion of corresponding previously received voice information should be triggered;

an event indicator indicating that processing of at least a portion of corresponding previously received voice information should be triggered; and

a condition indicator indicating that processing of at least a portion of corresponding previously received voice information should be triggered.